

DEC STD 070-10 Video Systems Reference Manual, Dynamically Redefinable Character Sets Extension

DOCUMENT IDENTIFIER: A-DS-EL00070-10-0000 Rev A, 04-Oct-1990

ABSTRACT: This document describes the interface to load and designate Dynamically Redefinable Character Sets (DRCS), which forms extensions to Levels 2, 3 and 4 of the Character Cell Display service classes and Levels 1 and 2 of the Hardcopy Display service classes.

APPLICABILITY: Mandatory for engineers designing hardware for terminal products and software engineers designing programs using terminal interfaces. Additional requirements are defined in the DEC STD 070-1 Concepts and Conformance Criteria.

STATUS: APPROVED 04-Oct-1990; use VTX SMC for current status.

This document is confidential and proprietary, and is the property of Digital Equipment Corporation. It is an unpublished work protected under the Federal copyright laws.

©Digital Equipment Corporation. 1990. All rights reserved.

DEC STD 070-10 Video Systems Reference Manual, Dynamically Redefinable Character Sets Extension

DOCUMENT IDENTIFIER: A-DS-EL00070-10-0000 Rev A, 04-Oct-1990

REVISION HISTORY:	Original Draft	30-Oct-1982
	Revision 0.1	25-Dec-1982
	Rev A,	04-Oct-1990

Document Management Category:	Terminal Interface Architecture (STI)
Responsible Department:	Video, Image, and Printing Systems (VIPS) Terminals Architecture
Responsible Person:	Peter Sichel

APPROVAL: This standard, prepared by the Desktop Systems Group, has been reviewed and recommended for approval by the General Review group for its category.

Peter Sichel, Video, Image, and Printing Systems

Eric Williams - Standards Process Manager

Direct requests for further information to:

Peter Sichel
Use \$ VTX ELF for the latest location information.

Use VTX SMC to order copies of this document from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

CONTENTS

1 INTRODUCTION	1
1.1 VIDEO CHARACTER CELL DISPLAY REQUIREMENTS	1
1.2 HARDCOPY CHARACTER CELL DISPLAY REQUIREMENTS	1
2 TERMINOLOGY	1
3 FUNCTIONAL DESCRIPTION	2
3.1 LOADING FONT FILES	2
3.2 ASSOCIATING FONT FILES WITH CHARACTER SETS	2
3.3 DESIGNATING AND INVOKING CHARACTER SETS	3
3.4 NOTE ON FUTURE USE OF FONTS	3
3.5 CELL MATRIX SIZE, SET SIZE, AND FONT USAGE	3
4 CONTROL FUNCTIONS	5
4.1 INTRODUCER SEQUENCE FORMAT	6
4.1.1 Parameter (Pfn): Font Number	6
4.1.2 Parameter (Pcn): First Character In Set To Load	6
4.1.3 Parameter (Pe): Erase Control	7
4.1.4 Parameter (Pcmw): Character Cell Matrix Width	7
4.1.5 Parameter (Pss): Font Set Size (Video Terminals Only)	8
4.1.6 Parameter (Pu): Font Usage	9
4.1.7 Parameter (Pcmh): Character Cell Matrix Height	9
4.1.8 Parameter (Pcss): Character Set Size	10
4.2 COMMAND STRING FORMAT	10
4.2.1 Character Set Designation String	10
4.2.2 Character Font Data	11
4.2.2.1 Summary	11
4.2.2.2 Sixel Definition	12
4.2.2.3 Sixel Bit Pattern	12
4.3 ALGORITHM	14
5 DRCS FONT LOADING EXAMPLES	20
5.1 EXAMPLE ONE	20
5.2 EXAMPLE TWO	20
5.3 EXAMPLE THREE	21
6 CHANGE HISTORY	22
6.1 Revision 0.0 To AX10	22
6.2 Rev AX10 to AX11	22
6.3 Rev AX11 to AX12	22
6.4 Rev AX12 to AX13	22
Appendix A REFERENCED DOCUMENTS	25

A.1 EL-Class Digital Documents 25

INDEX

1 INTRODUCTION

Dynamically Redefinable Character Sets (DRCS) are an extension to levels 2, 3 and 4 of the Character Cell service class and levels 1 and 2 of the Hardcopy Display service class. They provide the capability of loading into the terminal a font description which can be designated and invoked as a resident character set. The font loading format as defined for DRCS is flexible enough to be applicable to a wide variety of devices, including display and printing devices of various resolutions, and varying amounts of font storage capability.

In this document (DEC STD 070-10), the term "font file" refers to a sequence of bit patterns that define the graphic symbols of a coded character set.

1.1 VIDEO CHARACTER CELL DISPLAY REQUIREMENTS

In Level 2 only one 94-character font buffer is required for conformance. Because Level 2 is a Character Cell Display, the font files which can be loaded are fixed width only. A 94- or 96-character font buffer is required for conformance in Level 2 if the device supports a 96-character graphic set designation. The 96-character graphic set designation is part of the 8-bit Interface Architecture Extension. In Level 3 and 4, only one character font buffer is required to support either a 94- or 96-character set for conformance; support of 96-character graphic sets is required in Level 3 or higher. Refer to *DEC STD 070-3 Code Extension Layer* for more information.

For Level 2, 3 or 4 conforming devices, the minimum number of logical pixels per character in the DRCS font buffer is 80, defining a character cell matrix 8 wide by 10 high. This does not include inter-character space for text characters, which is provided automatically when text format is specified.

1.2 HARDCOPY CHARACTER CELL DISPLAY REQUIREMENTS

For devices which conform to Level 1 or 2 of the Hardcopy Display service class, a minimum of three character font buffers is required, each supporting either a 94- or 96-character graphic character set; support of 96-character graphic character sets is required in all levels of the Hardcopy Display service class. The number of logical pixels per character is device-dependent but shall support at least the resolution of the draft-quality resident font files. Hardcopy DRCS fonts may use the whole character cell, including inter-character space.

2 TERMINOLOGY

Cell Matrix Size - The width and height in logical pixels of a character cell.

Character Cell - A rectangular bit pattern which defines the foreground and background values of a single character rendition.

Coded Character Set - A set of named characters and their one-to-one relationship with bit combinations.

Control String - A class of control functions (bit strings) introduced by a special bit combination and terminated by a string terminator, the interpretation of which is dependent on the introductory character sequence.

DCS Introducer Sequence - The sequence of characters in a Device Control String which immediately follows the DCS control code and precedes the command string. Within Digital, DCS Introducer Sequences have the same format and follow the same parsing rules as control sequences.

Designate - To identify a set of characters that are to be represented, in some cases immediately and in others on the further occurrence of a control function, in a prescribed manner.

Font - An instance of a type family having a specific type size, style, characteristic weight, and character proportion.

Font File - A sequence of bit patterns or picture descriptors which define the graphic symbols of a coded character set.

Escape Sequence - A bit string that is for control purposes in code extension procedures consisting of two or more bit combinations, the first of which is an Escape (ESC) character.

Invoke - To cause a designated set of characters to be the prescribed bit combinations whenever those bit combinations occur, until an appropriate code extension function occurs.

Logical Pixel - A single element of a binary image, which may (but does not necessarily) map to a physical pixel in either the transmitting or receiving device.

Scan - One pass across an image in either the encoding or decoding process, consisting of the height of one sixel, or six pixels.

Set Size - A value which defines the relationship of a font definition to physical display parameters.

Sixel - A 7-bit or 8-bit character, the lower six bits of which represent binary image data.

3 FUNCTIONAL DESCRIPTION

3.1 LOADING FONT FILES

The DRCS font buffers may be down-line loaded on command from the host, using the Down-Line Load (DECDLD) control string. The DECDLD command string consists of a sequence of binary data in sixel format which defines the composition of the font file.

3.2 ASSOCIATING FONT FILES WITH CHARACTER SETS

The command string also contains parameters which specify the association between a character set designation string and the font file to be loaded into a buffer. The downloaded font file may be identified as replacing one of the character sets built into the device (for example, ASCII) or as a new character set not present in the device. Refer to *DEC STD 070-3 Code Extension Layer* and *DEC STD 070-5 Character Cell Display*.

When the character set designation string of a loaded font file which replaces a built-in character set is changed, either to another built-in set, or to a new character set, the built-in set that was replaced must be made available in the device without additional interchange.

It is recommended for new devices that when a loaded font file which replaces a built-in character set is completely erased by an application, via the Pe parameter, that the device make the original built-in set available without additional interchange (not mandatory).

3.3 DESIGNATING AND INVOKING CHARACTER SETS

Once a font has been loaded its associated character set, whether DRCS or otherwise, may be designated to any of the four G-sets (G0, G1, G2, or G3), and may invoked into either the left or right hand sides (GL or GR) of the In Use Table, according to the restrictions of ANSI code extension.

NOTE

96-character fonts may not be directly designated into G0. Refer to the "Code Extension Layer" section.

In other words, it may be used in any manner that standard character sets are used including, if applicable, as the User Preference Set.

3.4 NOTE ON FUTURE USE OF FONTS

It should be noted that the font file loading mechanism as defined in this specification for DRCS character sets may have broader application in the future, when fonts are used widely as a means of modifying the graphic rendition of a particular coded character set. In this case, the font number parameter included in the DECDLD load sequence will be used as a font identifier which may subsequently be assigned to a rendition value and selected using the Select Graphic Rendition (SGR) control function. The assignment may be accomplished using the Font Selection (FNT) control as defined in ANSI X3.64-1979 (p.37).

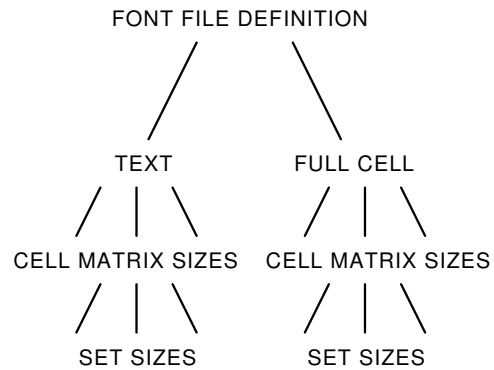
3.5 CELL MATRIX SIZE, SET SIZE, AND FONT USAGE

There are four parameters which affect the relationship between a font file definition as loaded into the font buffer memory and its physical appearance on the display. These parameters are:

1. Character Cell Matrix Size
2. Font Set Size
3. Font Usage

For any given font file, there are two possible Font Usage values, Text and Full Cell. For any given Font Usage, there are appropriate Character Cell Matrix Sizes which are typically device dependent. Furthermore, each Character Cell Matrix Size in a particular device may be limited to rendition in certain Set Sizes (80- or 132-column widths, for example).

The following diagram describes this relationship:



Because of the device dependent nature of these interactions, the product literature must be consulted for each device to determine the appropriate combination of parameters for a particular application.

4 CONTROL FUNCTIONS

DOWN LINE LOAD (FONT FILE)

DECDLD

Levels:

2X, 3X, 4X Video Character Cell
1X, 2X Hardcopy Display

Purpose: Down-line load font file patterns.

Format: The DECDLD control is a Device Control String consisting of an introducer sequence, followed by a command string. The format of the introducer sequence is:

```
DCS Pfn ; Pcn ; Pe ; Pcmw ; PSS ; Pu ; Pcmh ; PCSS {
9/0 Parameter String ... 7/11
```

The introducer sequence is terminated by the final character { (7/11).

The command string format is:

```
Dscs Sxbp1 ; Sxbp2 ; ... ; Sxbpn ST
```

The command string is terminated by the String Terminator character 9/12 (in 7-bits, ESC \, or 1/11 5/12). Refer to DEC STD 070-3 for a complete description of the Device Control String format.

Description:

The DECDLD control string down-line loads one or more characters of a specified 94-character or 96-character (if required) Dynamically Redefinable Character Set (DRCS) with a specified logical pixel pattern.

On video terminals, the affected characters in the DRCS set may also change the presentation of characters from this set currently being displayed on the screen. The DECDLD control does not clear the screen.

In printers, DECDLD will not affect the presentation of characters already on the page.

Notes:

1. Only character positions 2/1 through 7/14 may be loaded in 94-character sets.
2. Character positions from 2/0 through 7/15 may be loaded in 96-character sets.
3. Character cell sizes are given in logical pixels, width by height.
4. Loaded character sets are not required to be loaded into non-volatile memory. Therefore, if the terminal is powered down all loaded sets may be lost.
5. The host cannot read back loaded character sets.
6. The fonts loaded by the DECDLD command are separate and distinct from the ReGIS graphics "Load Pattern Cell" command. The ReGIS character sets may not be used except with ReGIS "text" commands.

7. A Soft Reset function (DECSTR) does not erase font files loaded by the DECDLD command (although the designation of character sets associated with those font files may change).
8. The soft character set may be cleared using any of four techniques: (1) Using the DECDLD function from the host; (2) By result of a power up test sequence; (3) From Setup - the RECALL or DEFAULT functions, (but not RESET TERMINAL); (4) On video terminals via the RIS sequence, ESC c, (but not DECSTR).
9. On video terminals, DECSCL performs a soft reset to put the terminal in a known presentation state without affecting data already displayed (does not clear DRCS). In printers, DECSCL is interpreted as "reset to base state" and will clear any DRCS.

4.1 INTRODUCER SEQUENCE FORMAT

The following selective parameters control the operation of the font load function. All parameters have the same syntax as a control sequence and may be omitted, in which case the default values are used.

4.1.1 Parameter (Pfn): Font Number

The font number parameter, Pfn, specifies the particular DRCS font buffer to be loaded, starting with buffer 1.

On video terminals, only one font buffer is required for conforming products implementing the DRCS extension. An omitted parameter, or a parameter value of zero, will default to the first empty font buffer, or, if none are empty, to font buffer 1.

On printers, three font buffers are required. An omitted parameter, or a parameter value of zero, will default to font buffer 1.

Attempts to load font buffers greater than the required number will produce results which are UNDEFINED by the architecture. Therefore, conforming software will not specify font numbers greater than the number of font buffers required. Hardware will ignore attempts to load font buffers greater than the number implemented.

4.1.2 Parameter (Pcn): First Character In Set To Load

The second parameter selects the first character in the font buffer to be loaded. Parameter value 0 means 2/0, 1 means 2/1, 2 means 2/2, ... 94 means 7/14, and 95 means 7/15.

If the device supports only 94-character buffers, (Level 2 conforming video products that do not include the 8-bit Interface Architecture Extension), attempts to start loading at location zero will cause the first character definition to be ignored, and the second character definition will be used for position 2/1. Any attempt to start loading beyond (or continue beyond) location 94 will also be ignored.

If the device supports 94- or 96-character buffers, (required for printers and Level 3 or higher conforming video products), the Pcsc parameter (see below) determines whether the font file is being loaded for a 94-character or 96-character set. If a 94-character set is being loaded, the restricted 94-character buffer interpretation of Pcn applies. If a 96-character set is being loaded, attempts to start loading at location zero will load the character 2/0, and loading at a location greater than 95 will be ignored.

Conforming software loading 94-character sets or font buffers, will not specify values less than one or greater than 94. Conforming software loading 96-character fonts will not specify values greater than 95.

4.1.3 Parameter (Pe): Erase Control

The third parameter controls which characters are erased before loading takes place according to:

- 0 - Erase all characters in the specified font buffer (default)
- 1 - Erase only the characters that are loaded
- 2 - Erase all characters in all font buffers

A parameter value of 1 permits the changing of one or more characters in the specified font without affecting the remaining characters of that font. Characters which have been erased from the buffer and not redefined are represented in the visual display by the error character (reverse question mark).

Guideline:

For video terminals, conforming software should only use the parameter value of 2 (erase all characters in all font buffers) in response to an explicit user request.

4.1.4 Parameter (Pcmw): Character Cell Matrix Width

The fourth parameter defines the limit of the character cell matrix width in pixels of the font being loaded. Note that legal values for this parameter are dependent on the values of the Font Set Size parameter and the Font Usage parameter, and may vary from device to device.

The legal combinations of these three parameters must be clearly defined in the product literature for conforming terminals. Illegal values of the Character Cell Matrix Width parameter cause the remainder of the load sequence to be ignored. Changes to this parameter since the last DECDLD sequence to this buffer will result in the erasure of the entire set, and cause a new load to start.

Guideline:

Implementations may discard pixels beyond the width of the character cell specified by this parameter. Software can use the Character Matrix Width parameter to obtain device independent, predictable results.

Additional Requirements for Video Terminals:

For video terminals, some values of the Character Cell Matrix Width parameter also specify the Character Cell Matrix Height, as shown in the table below. If one of these parameters is specified, the Character Cell Matrix Height (Pcmh) parameter is ignored.

Possible values of the parameter for conforming devices include:

- 0 - Device default
- 1 - Not used, illegal
- 2 - 5 x 10 matrix (width x height)
- 3 - 6 x 10 matrix (width x height)
- 4 - 7 x 10 matrix (width x height)

- 5 - 5 pixel wide matrix
- 6 - 6 pixel wide matrix
- 7 - 7 pixel wide matrix
- 8 - 8 pixel wide matrix
- 9 - 9 pixel wide matrix
- 10 - 10 pixel wide matrix
- 11 - 11 pixel wide matrix
- 12 - 12 pixel wide matrix

A parameter value of 0 indicates that the width (or matrix) of the character cell is the device default, which is dependent on the implementing device.

Guideline:

7 x 10 is the device default for Level 2 video devices. 5 x 10, 6 x 10 and 7 x 10 character cells are always scaled for VT220 compatibility.

4.1.5 Parameter (Pss): Font Set Size (Video Terminals Only)

This parameter specifies the intended visual rendition of the font file, and is used by the device when such restriction can affect the use of the loaded set. For example, some implementations restrict a loaded font file to use in an 80-column or 132-column screen. Note that legal values for this parameter are dependent on the values of the Character Cell Matrix Size parameter and the Font Usage parameter, and may vary from device to device.

The legal combinations of these three parameters must be clearly defined in the product literature for conforming terminals. Illegal values of the Font Set Size parameter cause the remainder of the load sequence to be ignored. Changes to this parameter since the last DECDLD sequence to this buffer will result in the erasure of the entire set, and cause a new load to start.

Possible values of the parameter for conforming devices include (VT420 Example):

- 0 - Device default (80 columns, 24 lines, not mandatory)
- 1 - 80 columns, 24 lines, (normal rendition)
- 2 - 132 columns, 24 lines, (normal rendition)
- 11 - 80 columns, 36 lines, (normal rendition)
- 12 - 132 columns, 36 lines, (normal rendition)
- 21 - 80 columns, 48 lines, (normal rendition)
- 22 - 132 columns, 48 lines, (normal rendition)

NOTE

The values for this parameter may be extended in the future to include character pitch and other physical rendition values which may affect the use of the loaded set.

4.1.6 Parameter (Pu): Font Usage

The sixth parameter defines the loaded font file as either a text font file or as a full cell font file. Full cell font files can individually address all pixels in a cell, while text font files may not be able to address all pixels individually. Text font files may also define the centering and spacing characteristics of the area around the character cell. Note that legal values for this parameter are dependent on the values of the Character Cell Matrix Size parameter and the Font Set Size parameter, and may vary from device to device.

The legal combinations of these three parameters must be clearly defined in the product literature for conforming terminals. Illegal values of the Font Usage parameter cause the remainder of the load sequence to be ignored. Changes to this parameter since the last DECDLD sequence to this buffer will result in the erasure of the entire set, and cause a new load to start.

Possible values of this parameter for conforming devices are:

- 0 - Device default (Guideline: text on VT220)
- 1 - Text
- 2 - Full Cell

4.1.7 Parameter (Pcmh): Character Cell Matrix Height

The seventh parameter defines the limit of the character cell matrix height in pixels of the font file being loaded. Note that legal values for this parameter are dependent on the values of the Font Set Size parameter and the Font Usage parameter, and may vary from device to device.

The legal combinations of these three parameters must be clearly defined in the product literature for conforming terminals. Illegal values of the Character Cell Matrix Height parameter cause the remainder of the load sequence to be ignored. Changes to this parameter since the last DECDLD sequence to this buffer will result in the erasure of the entire set, and cause a new load to start.

On video terminals, some values of the Character Cell Matrix Width (Pcmw) parameter also specify the Character Cell Matrix Height. If one of these Pcmw parameters is specified, the Character Cell Matrix Height (Pcmh) parameter is ignored.

The value of this parameter indicates the height of the Character Cell Matrix. For example, a value of 15 indicates a 15-pixel high character cell matrix. If a parameter value of 0 is specified, this indicates that the height of the character cell matrix is the device default, which is dependent on the implementing device.

Guideline:

Implementations may discard pixels beyond the height of the character cell specified by this parameter. Software can use the Character Matrix Width parameter to obtain device independent, predictable results.

4.1.8 Parameter (Pcss): Character Set Size

The eighth parameter controls the size of the associated character set (and modifies the interpretation of the Character Number (Pcn) parameter). The value of the parameter indicates the size of character set associated with the downloaded font according to:

- 0 - 94-character set (default)
- 1 - 96-character set

A parameter value of 1 permits the loading of the 2/0 and 7/15 characters in the font. And requires that the 96-character designating escape sequences (refer to the "Code Extension Layer" section) be used to designate the font.

The Final character { (7/11) indicates that this DCS string is the DECDLD control function. The data following it up to the String Terminator (ST) is from 2/0 to 7/14 and represents the actual font load command string.

4.2 COMMAND STRING FORMAT

The syntax of the DECDLD command string is as follows:

1. Dscs - (Select Character Set) The command string begins with a character set designation string, which defines the intermediate and final characters of the escape sequence which will subsequently be used to designate the loaded font as a character set.
2. Sxbr - The final character of the character set designation string is followed by a sequence of sixel bit pattern strings, each string containing the complete definition for one character of the font. Refer to subhead 10.4.2.2. Devices which support only 94-character sets should include only 94-sixel bit pattern strings in a single DECDLD command. Devices which support 96-character sets may include up to 96 sixel bit pattern strings in a single DLD command. Additional patterns will be ignored as necessary, but will not cause the command to be aborted.
3. ST - The String Terminator (ST, 9/12) control.

4.2.1 Character Set Designation String

The Dscs string is a set of zero, one, two, or three ANSI intermediates, followed by an ANSI final. (Dscs = I I I F). The Dscs defines the character set for the loaded font, and is used in the SCS (Select Character Set) escape sequence. Note that the same Dscs can be used to describe a 94-character or 96-character graphic set. The Pcss parameter indicates which initial intermediate character of the SCS sequence is to be used to designate the character set (see below). Refer to the DEC STD 070-3 and DEC STD 070-5 for a complete description of the escape sequence format, and particularly the SCS sequence.

The SCS sequence for designating the loaded font will be of the form:

- ESC (Dscs - Load the set into G0 (94-character set)
- ESC) Dscs - Load the set into G1 (94-character set)
- ESC * Dscs - Load the set into G2 (94-character set)
- ESC + Dscs - Load the set into G3 (94-character set)
- ESC - Dscs - Load the set into G1 (96-character set)
- ESC . Dscs - Load the set into G2 (96-character set)
- ESC / Dscs - Load the set into G3 (96-character set)

Changes to the Dscs string since the last DECDLD sequence to this buffer will result in the erasure of the entire set, and cause a new load to start.

Examples of Dscs:

1. Dscs = sp @
Defines the loaded font file to be a DRCS set. (This particular value is the recommended value for user defined sets for either 94-character or 96-character graphic character sets.)
2. Dscs = B
Defines the loaded font file to be ASCII (94-character), or ISO Latin Alphabet Nr 2 supplemental (96-character), depending on the value of the Pcsc parameter.
3. Dscs = & % C
Defines the loaded font file to an as yet unregistered set.

4.2.2 Character Font Data

4.2.2.1 Summary

The remainder of the command data is a sequence of sixel bit pattern strings (Sxsb string), each defining the pattern for a single character in the font. Each Sxsb string is separated by a semicolon character (;, 3/11). Each individual Sxsb string consists of several sequences of sixel data (refer to subhead 10.4.2.2.3) which define the individual horizontal scans of the character pattern, each sixel data sequence being terminated by a slash character (/ , 2/15).

Bit combinations 0/8 to 0/13 may be included in the command string, and have no affect on the interpretation of the data. They permit convenient formatting of command strings when being created by text editors since TAB, CR, and LF are included in this range. The following bit combinations are considered as error conditions when they occur within the character font data. Because existing implementations differ in how they handle these bit combinations, their processing is UNDEFINED by the architecture. It is not mandatory, but recommended that future implementations adopt the following conventions for these bit combinations.

- C0 controls 0/0 to 0/7 and 0/14 to 1/15 should be ignored except for ESC 1/11 which terminates the DCS and begins a new sequence. ESC followed immediately by \ 5/12 is the 7-bit representation of STRING TERMINATOR and ends the DECDLD normally. Other ESC sequences should cause the entire DECDLD command string to be ignored (aborted).
- 2/0 and 7/15 should be ignored.
- 2/1 to 3/14 with the exception of 2/15 and 3/11 should be ignored.
- C1 controls 8/0 to 9/15 should abort the DECDLD command string except for 9/12 STRING TERMINATOR which ends the DECDLD normally.
- 10/0 and 15/15 should be ignored.
- 10/1 to 15/14 should be interpreted as 2/1 to 7/14.

These characters should not be used by conforming software (except 2/15, 3/11, and STRING TERMINATOR).

4.2.2.2 Sixel Definition

A sixel is an encoded representation of six pixels. It is used to transfer pixel information in Device Control Strings. Sixel data consists of characters from ? (3/15) through (7/14), in which the character code represents a binary bit pattern. Each character represents a binary value plus 63 decimal. Thus 3/15 represents the binary value 000000, and 7/14 represents 111111. The bits are represented as a vertical, 1 by 6 pixel matrix ("the sixel"), with the least significant bit at the top. A reset bit (zero) specifies the font background, and a set bit (one) specifies the font foreground.

NOTE

Refer to *DEC STD 070-9 Sixel Graphics Extension* for a more complete description of the conversion algorithms. Note that the DECDLD command string uses the sixel data format, but does not use the control functions as defined in the Sixel Graphics Extension.

4.2.2.3 Sixel Bit Pattern

To load the logical pixel pattern for an individual character in a font (one Sxbp), a sequence of sixel characters is sent, with each sixel specifying one vertical column of six logical pixels. After each character the font pattern scan advances horizontally by one logical pixel. This sequence continues until either a scan terminator (the / character, 2/15) is received, a sixel bit pattern terminator (the ; character, 3/11) is received, or a STRING TERMINATOR (9/12) is received. If the number of characters in a horizontal scan is greater than the width of the loaded character (as defined in the Character Cell Matrix Size parameter of the introducer sequence), the characters in excess of the loaded character width are ignored.

The scan terminator character (/ , 2/15) advances the font pattern scan to the next lower row of six vertical logical pixels. When the height of a loaded character (as defined in the Character Cell Matrix Size parameter of the introducer sequence) is not a multiple of six, the high order bits of the sixel values are ignored on the last row of the font pattern. If the number of horizontal scans is greater than the height of the loaded character (divided by six), the characters in excess of the loaded character height are ignored.

Any pixels of a loaded character that are not specified are set to zero. This might occur either as a result of a horizontal scan being terminated before the pattern has reached the full width of the loaded character, or as the result of the sixel bit pattern string being terminated before the full character matrix has been defined.

The following picture shows the mapping of sixels onto an 8 x 10 logical pixel character cell:

1	2	3	4	5	6	7	8
b0	b0	b0	b0	b0	b0	b0	b0
b1	b1	b1	b1	b1	b1	b1	b1
b2	b2	b2	b2	b2	b2	b2	b2
b3	b3	b3	b3	b3	b3	b3	b3
b4	b4	b4	b4	b4	b4	b4	b4
b5	b5	b5	b5	b5	b5	b5	b5
b0	b0	b0	b0	b0	b0	b0	b0
b1	b1	b1	b1	b1	b1	b1	b1
b2	b2	b2	b2	b2	b2	b2	b2
b3	b3	b3	b3	b3	b3	b3	b3
9	10	11	12	13	14	15	16

The rows of numbers on the top and bottom relate to the order in which sixel values are sent to the terminal.

The following picture shows the mapping of Sixels onto an 10 x 20 logical pixel full-cell character cell:

1	2	3	4	5	6	7	8	9	10
b0	b0	b0	b0	b0	b0	b0	b0	b0	b0
b1	b1	b1	b1	b1	b1	b1	b1	b1	b1
b2	b2	b2	b2	b2	b2	b2	b2	b2	b2
b3	b3	b3	b3	b3	b3	b3	b3	b3	b3
b4	b4	b4	b4	b4	b4	b4	b4	b4	b4
b5	b5	b5	b5	b5	b5	b5	b5	b5	b5
11	12	13	14	15	16	17	18	19	20
b0	b0	b0	b0	b0	b0	b0	b0	b0	b0
b1	b1	b1	b1	b1	b1	b1	b1	b1	b1
b2	b2	b2	b2	b2	b2	b2	b2	b2	b2
b3	b3	b3	b3	b3	b3	b3	b3	b3	b3
b4	b4	b4	b4	b4	b4	b4	b4	b4	b4
b5	b5	b5	b5	b5	b5	b5	b5	b5	b5
21	22	23	24	25	26	27	28	29	30
b0	b0	b0	b0	b0	b0	b0	b0	b0	b0
b1	b1	b1	b1	b1	b1	b1	b1	b1	b1
b2	b2	b2	b2	b2	b2	b2	b2	b2	b2
b3	b3	b3	b3	b3	b3	b3	b3	b3	b3
b4	b4	b4	b4	b4	b4	b4	b4	b4	b4
b5	b5	b5	b5	b5	b5	b5	b5	b5	b5
31	32	33	34	35	36	37	38	39	40
b0	b0	b0	b0	b0	b0	b0	b0	b0	b0
b1	b1	b1	b1	b1	b1	b1	b1	b1	b1

The rows of numbers on the top and bottom relate to the order in which sixel values are sent to the terminal.

4.3 ALGORITHM

State Affected:

```

FONT: ARRAY [1..MAX_NUM_FONTS] OF FONT_BUFFER_TYPE;
CELL_NUMBER: CHARACTER_VALUE_TYPE;
FONT_NUMBER: 1..MAX_NUM_FONTS;
FIRST_CHARACTER: CHARACTER_VALUE_TYPE;
ERASE_CONTROL: (ERASE_THIS_FONT,NO_ERASE,ERASE_ALL_FONTS);
CELL_WIDTH: CELL_WIDTH_TYPE;
CELL_HEIGHT: CELL_HEIGHT_TYPE;
SET_SIZE: (EIGHTY,ONE_THIRTY_TWO);
CELL_USAGE: (TEXT,FULL_CELL);
CHARACTER_SET_SIZE: (94,96);
CHARACTER_SET_DESIGNATOR:
    ARRAY [1..MAX_LENGTH_DESIGNATOR] OF
        DESIGNATOR_CHARACTER_TYPE;

```

Algorithm:

```

CONST    MAX_NUM_FONTS = 1;

        MIN_CHARACTER_VALUE = 0;
        MAX_CHARACTER_VALUE = 95;
        (* Devices which are not required to support
           96-character sets may set these values to 1 and 94 *)

        MAX_CELL_WIDTH = 8;
        MAX_CELL_HEIGHT = 10;
        (* These values may vary from device to device *)

        MAX_LENGTH_DESIGNATOR = 3;

        ON = TRUE;
        OFF = FALSE;

TYPE     CELL_WIDTH_TYPE = 1..MAX_CELL_WIDTH;
        CELL_HEIGHT_TYPE = 1..MAX_CELL_HEIGHT;

        CHARACTER_CELL_TYPE =
            ARRAY [CELL_WIDTH_TYPE,CELL_HEIGHT_TYPE] OF BOOLEAN;

        CHARACTER_VALUE_TYPE =
            MIN_CHARACTER_VALUE..MAX_CHARACTER_VALUE;

        FONT_BUFFER_TYPE =
            ARRAY [CHARACTER_VALUE_TYPE] OF CHARACTER_CELL_TYPE;

VAR      FONT: ARRAY [1..MAX_NUM_FONTS] OF FONT_BUFFER_TYPE;
        CELL_NUMBER: CHARACTER_VALUE_TYPE;
        FONT_NUMBER: 1..MAX_NUM_FONTS;
        FIRST_CHARACTER: CHARACTER_VALUE_TYPE;
        DRCS_ERASE_CONTROL:
            (ERASE_THIS_FONT,NO_ERASE,ERASE_ALL_FONTS);
        CELL_WIDTH: CELL_WIDTH_TYPE;
        CELL_HEIGHT: CELL_HEIGHT_TYPE;
        SET_SIZE: (EIGHTY,ONE_THIRTY_TWO);
        CELL_USAGE: (TEXT,FULL_CELL);
        CHARACTER_SET_SIZE: (94,96);
        VALID_DRCS_SYNTAX: BOOLEAN;
        DRCS_STRING_TERMINATED: BOOLEAN;
        H,V,N: INTEGER;

        PROCEDURE GET_NEXT_EVENT (VAR EVENT: EVENT_BUFFER_TYPE); EXTERN;
        PROCEDURE GET_DRCS_PARAMETERS (EVENT: EVENT_BUFFER_TYPE); FORWARD;
        PROCEDURE GET_CHARACTER_SET_DESIGNATOR; FORWARD;
        PROCEDURE FILL_CELLS; FORWARD;

```

```

PROCEDURE EXECUTE_FONT_LOAD (EVENT: EVENT_BUFFER_TYPE);
(*
  This routine processes DECDLD font load command strings.
  The input to this routine is the parsed DCS introducer
  sequence. Successive calls are made to the ANSI parser
  (GET_NEXT_EVENT) to input a character at a time for
  processing. The routine will not exit until a String
  Terminator (9/12) is received.
*)
BEGIN
  (* first, process the DCS introducer sequence parameters *)
  GET_DRCS_PARAMETERS (EVENT);
  IF VALID_DRCS_SYNTAX THEN
    BEGIN
      DRCS_STRING_TERMINATED:= FALSE;
      (* if the sequence is valid, get the character set designator *)
      GET_CHARACTER_SET_DESIGNATOR;
      IF VALID_DRCS_SYNTAX THEN
        BEGIN
          (* erase according to the erase parameter *)
          CASE DRCS_ERASE_CONTROL OF
            ERASE_THIS_FONT:
              FOR CELL_NUMBER:=
                MIN_CHARACTER_VALUE TO MAX_CHARACTER_VALUE DO
                FOR H:= 1 TO MAX_CELL_WIDTH DO
                FOR V:= 1 TO MAX_CELL_HEIGHT DO
                FONT[FONT_NUMBER,CELL_NUMBER,H,V]:= OFF;
            ERASE_ALL_FONTS:
              FOR N:= 1 TO MAX_NUM_FONTS DO
                FOR CELL_NUMBER:=
                MIN_CHARACTER_VALUE TO MAX_CHARACTER_VALUE DO
                FOR H:= 1 TO MAX_CELL_WIDTH DO
                FOR V:= 1 TO MAX_CELL_HEIGHT DO
                FONT[FONT_NUMBER,CELL_NUMBER,H,V]:= OFF;
            OTHERWISE IGNORE;
          END;
          (* then process the command string *)
          FILL_CELLS;
          (* on overflow, keep going until String Terminator *)
          WHILE NOT DRCS_STRING_TERMINATED DO
            BEGIN
              GET_NEXT_EVENT (EVENT);
              IF (EVENT.EVENT_CODE = CONTROL_CODE)
                AND (EVENT.CODE_VALUE = 156) THEN
                DRCS_STRING_TERMINATED:= TRUE;
            END;
          END;
        END;
      END;
    END;
  END;
END;

```

```

PROCEDURE GET_DRCS_PARAMETERS;
(*
  This routine gets the DCS introducer sequence parameters
  from the event buffer and sets the appropriate state
  prior to loading the font.
*)
BEGIN
VALID_DRCS_SYNTAX:= TRUE;
(* the first parameter is the font number to be loaded *)
IF (EVENT.PARAMETERS[1] = 0) OR (EVENT.PARAMETERS[1] = 1) THEN
  FONT_NUMBER:= 1
ELSE VALID_DRCS_SYNTAX:= FALSE;
(* the second parameter is the first character to be loaded *)
FIRST_CHARACTER:= EVENT.PARAMETERS[2];
IF (FIRST_CHARACTER = 0) OR (FIRST_CHARACTER > 95) THEN
  FIRST_CHARACTER:= 1;
(* the third parameter is the erase control specifier *)
CASE EVENT.PARAMETERS[3] OF
  0: DRCS_ERASE_CONTROL:= ERASE_THIS_FONT;
  1: DRCS_ERASE_CONTROL:= NO_ERASE;
  2: DRCS_ERASE_CONTROL:= ERASE_ALL_FONTS;
  OTHERWISE VALID_DRCS_SYNTAX:= FALSE;
END;
(* the fourth parameter is the cell width or size *)
CELL_HEIGHT:= MAX_CELL_HEIGHT;      (* assume no cell height
                                     parameter at start *)
CASE EVENT.PARAMETERS[4] OF
  0: CELL_WIDTH:= MAX_CELL_WIDTH;    (* 0 is device default *)
  1: VALID_DRCS_SYNTAX:= FALSE;      (* 1 is illegal*)
  2: CELL_WIDTH:= 5;
  3: CELL_WIDTH:= 6;
  4: CELL_WIDTH:= 7;
  OTHERWISE CELL_WIDTH:= EVENT.PARAMETERS[4];
END;
IF (CELL_WIDTH > MAX_CELL_WIDTH)
  THEN CELL_WIDTH:= MAX_CELL_WIDTH;
(* the fifth parameter is the set size *)
CASE EVENT.PARAMETERS[5] OF
  0,1: SET_SIZE:= EIGHTY;
  2: SET_SIZE:= ONE_THIRTY_TWO;
  OTHERWISE VALID_DRCS_SYNTAX:= FALSE;
END;
(* the sixth parameter is the font usage *)
CASE EVENT.PARAMETERS[6] OF
  0,1: CELL_USAGE:= TEXT;
  2: CELL_USAGE:= FULL_CELL;
  OTHERWISE VALID_DRCS_SYNTAX:= FALSE;
END;
(* the seventh parameter (if present) is the cell height *)
CASE EVENT.PARAMETERS[7] OF
  0: CELL_HEIGHT:= MAX_CELL_HEIGHT;  (* device dependant default
*)
  OTHERWISE CELL_HEIGHT:= EVENT.PARAMETERS[7];
END;
IF (CELL_HEIGHT > MAX_CELL_HEIGHT)
  THEN CELL_HEIGHT:= MAX_CELL_HEIGHT;
(* the eighth parameter is the character set size *)
CASE EVENT.PARAMETERS[8] OF
  0: CHARACTER_SET_SIZE:= 94;
  1: CHARACTER_SET_SIZE:= 96;
  OTHERWISE VALID_DRCS_SYNTAX:= FALSE;
END;
END;

```

```

PROCEDURE GET_CHARACTER_SET_DESIGNATOR;
(*
  This routine parses the character set designating escape
  sequence to be associated with the loaded font. This sequence
  must immediately follow the DCS introducer sequence.

  Refer to DEC STD 070-5 for a description of the character set
  table and the subsequent designation procedure.
*)
BEGIN
  (* accept characters until final or String Terminator *)
  N:= 1;
  REPEAT
    GET_NEXT_EVENT (EVENT);
    CASE EVENT.EVENT_CODE OF
      GRAPHIC_CODE:      (* put legal characters in the buffer *)
        BEGIN
          IF (N=1) AND (EVENT.CODE_VALUE >= 32)
            AND (EVENT.CODE_VALUE <= 47) THEN
            CHARACTER_SET_TABLE[4].FIRST_INTERMEDIATE:=
              EVENT.CODE_VALUE;
          IF (N=2) AND (EVENT.CODE_VALUE >= 32)
            AND (EVENT.CODE_VALUE <= 47) THEN
            CHARACTER_SET_TABLE[4].SECOND_INTERMEDIATE:=
              EVENT.CODE_VALUE;
          IF (EVENT.CODE_VALUE >= 48)
            AND (EVENT.CODE_VALUE <= 126) THEN
            CHARACTER_SET_TABLE[4].FINAL:= EVENT.CODE_VALUE;
          IF (EVENT.CODE_VALUE >= 32)
            AND (EVENT.CODE_VALE <= 126) THEN N:= N+1;
          END;
        CONTROL_CODE:    (* terminate on String Terminator, 9/12 *)
          IF EVENT.CODE_VALUE = 156 THEN DRCS_STRING_TERMINATED:= TRUE
          ELSE IGNORE;    (* ignore other control characters *)
          OTHERWISE IGNORE; (* ignore escape and control sequences *)
        END;
    UNTIL ((EVENT.EVENT_CODE = GRAPHIC_CODE)
    AND ((EVENT.CODE_VALUE >= 64) AND (EVENT.CODE_VALUE <= 126)))
    OR DRCS_STRING_TERMINATED;
  END;

```

```

PROCEDURE FILL_CELLS;
(*
  This routine processes the sixel command string and loads
  the character cells in succession, starting with the first
  character specified in the introducer sequence.
*)
VAR   NEXT_CELL: BOOLEAN;
BEGIN
  (* start with the first character as specified *)
  CELL_NUMBER:= FIRST_CHARACTER;
  NEXT_CELL:= TRUE;
  REPEAT
    (* clear the cell if not already cleared *)
    IF NEXT_CELL THEN
      BEGIN
        IF DRCS_ERASE_CONTROL = NO_ERASE THEN
          FOR H:= 1 TO MAX_CELL_WIDTH DO
            FOR V:= 1 TO MAX_CELL_HEIGHT DO
              FONT[FONT_NUMBER,CELL_NUMBER,H,V] := OFF;
            H:= 1;
            V:= 1;
            NEXT_CELL:= FALSE;
          END;
        (* now, fill it *)
        GET_NEXT_EVENT (EVENT);
        CASE EVENT.EVENT_CODE OF
          GRAPHIC_CODE:
            BEGIN
              (* convert sixels into the font memory *)
              IF (EVENT.CODE_VALUE >= 63)
              AND (EVENT.CODE_VALUE <= 126) THEN
                BEGIN
                  EVENT.CODE_VALUE:= EVENT.CODE_VALUE - 63;
                  IF H <= CELL_WIDTH THEN
                    BEGIN
                      FOR N:= 1 TO 6 DO
                        BEGIN
                          IF V+N-1 <= CELL_HEIGHT THEN
                            BEGIN
                              IF ODD(EVENT.CODE_VALUE) THEN
                                FONT[FONT_NUMBER,CELL_NUMBER,H,V+N-1] := ON;
                                (* shift in one bit at a time *)
                                EVENT.CODE_VALUE:= EVENT.CODE_VALUE DIV 2;
                              END
                            ELSE IGNORE;
                          END;
                          H:= H+1;
                        END
                      ELSE IGNORE;
                    END
                  ELSE
                    BEGIN
                      CASE EVENT.CODE_VALUE OF
                        (* / terminates a horizontal scan *)
                        47: BEGIN H:= 1; V:= V+6; END;
                        59: (* ; terminates a cell, go to the next *)
                          BEGIN
                            NEXT_CELL:= TRUE;
                            CELL_NUMBER:= CELL_NUMBER+1;
                            H:= 1; V:= 1;
                          END;
                        OTHERWISE IGNORE;
                      END;
                    END;
                  END;
                END;
              END;
            END;
          ELSE
            BEGIN
              CASE EVENT.CODE_VALUE OF
                (* / terminates a horizontal scan *)
                47: BEGIN H:= 1; V:= V+6; END;
                59: (* ; terminates a cell, go to the next *)
                  BEGIN
                    NEXT_CELL:= TRUE;
                    CELL_NUMBER:= CELL_NUMBER+1;
                    H:= 1; V:= 1;
                  END;
                OTHERWISE IGNORE;
              END;
            END;
          END;
        END;
      END;
    END;
  END;
END;

```

```
CONTROL_CODE:      (* terminate on String Terminator, 9/12 *)
  IF EVENT.CODE_VALUE = 156 THEN DRCS_STRING_TERMINATED:= TRUE
  ELSE IGNORE;      (* ignore other control characters *)
  OTHERWISE IGNORE; (* ignore escape and control sequences *)
END;
UNTIL (CELL_NUMBER > MAX_CHARACTER_VALUE)
OR (DRCS_STRING_TERMINATED);
END;
```


5. B is the designator for the ASCII character set - therefore, this font replaces ASCII, and may be designated using the following Select Character Set (SCS) sequences:
 ESC (B
 ESC) B
 ESC * B
 ESC + B
6. The first character loaded is blank.
7. The second character loaded is solid.
8. The third character loaded is top 6/10 blank, bottom 4/10 solid.
9. The fourth character loaded is a thick vertical bar.

5.3 EXAMPLE THREE

```
DCS 1 ; 65 ; 1 ; 4 ; 2 ; 2 ; 16 ; 1 { B
???/??????????/?????;
~~~~~ / ~~~~~ / ~~~~~;
?????/{{{{/~~~~~ /?????;
?~?/?~?/?~?
ST
```

Notes:

1. The font is a 132-column, full-cell font with character matrix size of 4 x 16
2. The first character loaded corresponds to the 6/0 position in the code table.
3. The first character loaded is blank
4. The second character loaded is solid
5. The third character loaded has the top half blank, and the bottom half solid
6. The fourth character is a thick vertical bar
7. The excess Sixel information (outside of the 4 x 16 cell) is ignored
8. "B" could be confused with the ASCII designator, however, since Pcsc is 1, making this a 96-character soft character set, this font does not replace the ASCII font. (It is, however, the registered designator for the supplemental character set of ISO Latin Alphabet Nr 2, and would replace that set, if it were present in the device.)
9. The character set can be designated using the 96-character SCS sequence. The Dscs is "B". The SCS sequences are:
 ESC - B
 ESC . B
 ESC / B

6 CHANGE HISTORY

6.1 Revision 0.0 To AX10

1. Added Introduction, Terminology, and Functional Description sections.
2. Revised the DECDLD command format.
3. Restricted the legal values for the Font Number parameter.
4. Restricted the legal values for the First Character parameter.
5. Added a parameter value to erase all font definitions (User Preference Feature).
6. Added parameters for Character Cell Matrix Size, Font Set Size, and Font Usage.
7. Extensively revised the section on Command String Format.
8. Added the DECDLD algorithm.
9. Added examples for DECDLD.
10. Added references to the Character Cell Display section for character set tables of designating escape sequences.

6.2 Rev AX10 to AX11

1. Added statement that starting to load at location greater than 94 will be ignored in level 2.

6.3 Rev AX11 to AX12

1. Added support and Character Set Size parameter for 96-character graphic character sets for Level 2 and 3 DRCS.
2. Modified meaning of Character Matrix parameter to mean Character Matrix Width parameter as a modification to Level 2 DRCS.
3. Added Character Matrix Height parameter as a modification to Level 2 DRCS.
4. Added example containing new parameters.
5. Extended behavior of Font Number parameter when parameter is omitted or zero.
6. Clarified behavior of DRCS which replace standard character sets when the DRCS is erased.
7. Updated algorithm to reflect new parameters and 96-character graphic character set support.

6.4 Rev AX12 to AX13

1. Changed title to "Video & Printer Systems Reference Manual".
2. Clarified use of terms "font", "font file", and "character set" to be more consistent with printer terminology. "Font file" refers to a sequence of bit patterns that define the graphic symbols of a coded character set. "Fonts" are styles (an instance of a type family).

3. Added minimum requirements for Hardcopy Display service class (three 96 character buffers).
4. Clarified that DRCS can be used as the User Preference Supplemental Set.
5. Clarified effect of DECDLD on characters already on the page in printers (do not change).
6. Specified behavior of DECSTR, RIS, and DECSCL with respect to DRCS buffers for both video and hardcopy devices.
7. Clarified differences in behavior of DECDLD parameters for video and hardcopy devices (Pfn, Pcmw, Pss, Pcmh). Printers have three buffers minimum. Printers never deduce Pcmh from Pcmw. Pss (Font set size) is for video terminals only. Extended list of possible values for Pss to reflect VT420 implementation.
8. Specified Dscs may include up to 3 intermediates in addition to the first SCS intermediate which specifies the G-set.
9. Included error handling recommendations for unexpected bit combinations. ESC and C1 controls abort DECDLD (except ST and DCS). 10/1 to 15/14 should be interpreted as 2/1 to 7/14. This reflects latest consensus among hardcopy and video implementations.
10. Clarified that ST may terminate an individual character load.

APPENDIX A REFERENCED DOCUMENTS

A.1 EL-Class Digital Documents

EL-Class Number	Document Title
EL-00070-01	<i>DEC STD 070-1 Video Systems Reference Manual - Concepts and Conformance Criteria</i>
EL-00070-03	<i>DEC STD 070-3 Video Systems Standard - Code Extension Layer</i>
EL-00070-05	<i>DEC STD 070-5 Video Systems Reference Manual - Character Cell Display</i>
EL-00070-09	<i>DEC STD 070-9 Video and Printer Systems Reference Manual - SIXEL Graphics Extension</i>

Use VTX SMC to order copies of EL-class documents from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

INDEX

C

Cell Matrix
 width, 7
 Cell Matrix Size, 1, 3
 Character Cell, 1
 Character Cell Matrix Size, 8, 12
 Character Font Data, 11
 Coded Character Set, 1
 Conforming software, 7
 Control String, 1

D

DCS Introducer Sequence, 2
 DECDLD, 2, 3, 5, 7
 Designate, 2
 Down-Line Load, 2, 5
 DRCS, 1
 Dynamically Redefinable Character Set, 5
 Dynamically Redefinable Character Sets, 1

E

Erase Control, 7
 Escape Sequence, 2

F

FNT, 3
 Font, 2
 Font File, 2
 Font Number, 6
 Font Selection, 3
 Font Set Size, 3

I

Invoke, 2

L

Logical Pixel, 2

S

Scan, 2
 Select Graphic Rendition, 3
 Set Size, 2
 SGR, 3
 Sixel, 2
 bit pattern, 12
 definition, 12
 Soft Reset, 5

READER COMMENTS
Your comments and suggestions will help Standards and Methods Control improve their services and documents.

Did you request this document? _____ If so, did it arrive within a satisfactory period of time? _____ Please comment.

What are your impressions of this document? Consider format, organization, completeness, readability, and illustrations.

_____ **FOLD ON THIS LINE** _____

Did you find technical or clerical errors in this document? If so, please specify the page number(s) and the error(s).

Are the instructions for the update package clear? _____
Was an index available? _____ If not, is one needed? _____
Do you have other suggestions for improving this document?

The following information is optional:

Name _____ Mailstop _____
Department _____ Node _____

Send your comments to JOKUR::PROJECTS, or fold, staple, and send this page through interoffice mail to:

READERS' COMMENTS
STANDARDS AND METHODS CONTROL
NRO4/D4